Coverage-Guided Road Selection and Prioritization for Efficient Testing in Autonomous Driving Systems

Qurban Ali
University of Milano-Bicocca
Milan, Italy
q.ali@campus.unimib.it

Andrea Stocco
Technical University of Munich, fortiss GmbH
Munich, Germany
andrea.stocco@tum.de, stocco@fortiss.org

Leonardo Mariani
University of Milano-Bicocca
Milan, Italy
leonardo.mariani@unimib.it

Oliviero Riganelli
University of Milano-Bicocca
Milan, Italy
oliviero.riganelli@unimib.it

***Abstract*—Autonomous Driving Assistance Systems (ADAS) rely on extensive testing to ensure safety and reliability, yet road scenario datasets often contain redundant cases that slow down the testing process without improving fault detection. We present a novel test prioritization framework that reduces redundancy while preserving geometric and behavioral diversity. Road scenarios are segmented into representative sections, which are compared using similarity scores based on dynamic time warping and enriched with dynamic features of the ADAS driving behavior. These features guide clustering to identify groups of similar scenarios, from which representative cases are selected to guarantee coverage. Finally, we introduce a prioritization mechanism that ranks roads based on geometric complexity, driving difficulty, and historical failures, ensuring that the most critical and challenging tests are executed first. We evaluate our framework on the OPENCAT dataset and the Udacity self-driving car simulator using two ADAS models. On average, our approach achieves an 89% reduction in test suite size while retaining an average of 79% of failed road scenarios. The prioritization strategy improves early failure detection by up to 95× compared to random baselines. These results demonstrate that our framework significantly improves test efficiency and fault detection capability, while maintaining scenario diversity and generalizing across different ADAS.**

***Index Terms*—Autonomous Driving, Test Prioritization, Deep Learning Testing.**

I. Introduction

The validation of autonomous driving assistance systems (ADAS) is premised on the ability to expose failure-inducing behaviors before deployment on public roads [1]. Modern ADAS pipelines rely heavily on large-scale simulation or log replay, where thousands of kilometers of driving data are executed to evaluate the robustness of a driving policy under diverse road conditions [1], [2], [3], [4]. These datasets encompass a wide variety of road topologies, each carrying the potential to trigger critical misbehavior. However, the assumption that more data inherently leads to better validation has created a serious bottleneck: exhaustive replay is computationally expensive, frequently redundant, and offers no guarantee that high-risk cases will be encountered early. As ADAS models scale and regulations demand stronger safety evidence [5], indiscriminate road execution becomes both impractical and ineffective.

This paper focuses on regression testing of ADAS software, with a particular emphasis on test prioritization. Our work builds on the key observation that, in road-based testing, not all test cases carry the same value. Roads that appear visually distinct may still include segments that induce nearly identical driving behavior, offering no new insight into ADAS performance. Existing regression practices [6], [7] overlook these fine-grained redundancies, as they evaluate entire roads as whole test cases rather than analyzing behavioral variation within them. Furthermore, road execution is often performed in an arbitrary order or guided only by static topology similarity, without considering ADAS behavioral factors. As a result, critical failures are frequently discovered late, while early testing time is spent replaying roads containing benign or redundant segments.

In this work, we introduce a behavior-aware road prioritization framework for ADAS testing, which selects and reorders road execution based on how challenging each segment is for the driving model. In a nutshell, instead of treating all roads equally, our framework directs testing toward those segments most likely to expose failures. Our framework operates on existing datasets, constructing a behavioral profile for each road derived from trajectory-level signals such as steering variability, oscillatory corrections, cross-track error, and historical instability. Based on these behavioral profiles, we cluster redundant roads, eliminate those offering no novel insight, and rank the remaining segments according to their likelihood of revealing failures. A test case is considered a failure if the car goes out of bounds (OOB), i.e., off the road [8]. We validate our framework using state-of-the-art imitation learning ADAS across multiple driving environments, capturing diverse road geometries and driving dynamics. The evaluation shows that our prioritization framework consistently identifies the first failure significantly earlier than baseline strategies, including random replay and geometry-based ordering. In several environments, the first critical failure was discovered up to 95× times faster, and overall execution effort was reduced by more than half, without affecting failure exposure.

Our study highlights a broader implication for ADAS regression testing: evaluation should be driven by behavioral value rather than only dataset completeness. By focusing on behavioral difficulty, developers can accelerate risk discovery, reduce computational effort, and produce stronger evidence of system safety. The contributions of this paper are as follows:

Testing Framework. We introduce a novel behavior-aware framework for prioritizing road-based tests in autonomous driving, moving beyond structural diversity to behavioral challenge. We propose a redundancy filtering method to identify behaviorally equivalent roads and eliminate low-value replay without sacrificing failure exposure.

Evaluation. We conduct a comprehensive empirical evaluation showing substantial improvements in time-to-failure and testing efficiency over conventional replay strategies.

Replication. We provide a replication package containing our implementation and experiment scripts for both case studies to support reproducibility [9].

II. Background

A. ADAS Testing

Autonomous driving assistance systems depend on extensive testing to ensure safe and reliable behavior across diverse real-world conditions [10], [1]. Unlike conventional software, where correctness is verified through deterministic inputs and outputs, ADAS behavior emerges from continuous interaction with road geometry, vehicle dynamics, and control policies [11]. Testing must therefore account for dynamic responses such as steering corrections, lane keeping, braking patterns, and the vehicle's ability to remain stable under different conditions [12], [13].

In modern development pipelines, ADAS testing relies heavily on simulation and large-scale road log replay [14], [15], [16]. These road datasets are designed to cover a wide spectrum of driving environments. However, exhaustive replay of all available roads quickly becomes infeasible as datasets grow in size. Many roads exhibit redundant behavioral responses, offering minimal new insight into ADAS capabilities. To this aim, test prioritization is essential, as it enables the early execution of tests that are most likely to expose failures.

B. Test Prioritization

Test prioritization is traditionally concerned with ordering test cases so that those most likely to reveal faults are executed earlier [17]. In classical software engineering, prioritization strategies often rely on coverage metrics, change history, or past failures to rank tests. The goal is not to eliminate tests, but to improve the efficiency of fault discovery under resource constraints. Early execution of high-value tests reduces time-to-detection and accelerates developer feedback, which is critical in iterative deployment cycles.

Transferring this principle to ADAS introduces unique challenges [6]. Road tests are not isolated function calls; they represent continuous trajectories over time, influenced by control dynamics rather than discrete logic. As a result, conventional source-based prioritization techniques, such as statement coverage or mutation impact, are not applicable as they fail to capture what makes a road test valuable [11]. Instead, value must be derived from behavioral evidence: a road segment that induces high steering oscillation or lateral instability carries more fault potential than a visually complex but behaviorally trivial one [18].

We conjecture that effective prioritization for ADAS requires a shift from visual or input-based diversity toward behavior-based selection. Roads should not be distinguished by how different they appear as a whole, but by how different segments influence the driving model. This reframes prioritization around driving challenge, using fine-grained metrics [18] that capture difficulty, instability, or the potential for misbehavior, more effectively than uniformly replaying the entire dataset, or whole test cases.

III. Framework

Our framework includes a pipeline to extract, compare, cluster, and prioritize road sections (called *segments*) from existing whole tests, aiming to reduce fine-grained redundant test scenarios while preserving the geometric and behavioral diversity of road networks.

As illustrated in Figure 1, it starts with curvature-based road segmentation, decomposing roads into different section types (e.g., straight and curved) based on their geometric features. After that, it performs geometric matching using Dynamic Time Warping (DTW) [19] to identify repeated patterns across roads while enforcing curvature consistency (left vs right). In addition to geometric features, our framework leverages dynamic driving data (e.g., speed, steering, yaw rate, cross-track error) from simulation logs that supplement the geometric data, enabling more informative and realistic clustering of road sections.

These combined features jointly drive agglomerative clustering, which groups similar curves with similar ADAS behavior into compact clusters while preserving unique geometries. Once clusters have been obtained, the next step involves coverage-based road selection, which identifies a minimal subset of roads that collectively cover all clusters. This ensures rare or unique patterns are included while minimizing fine-grained redundancies.

Finally, our framework applies a prioritization mechanism that ranks both selected and unselected roads. The selected roads are assigned higher priority and ranked based on geometric diversity, dynamic difficulty, and historical failure data, ensuring that the most valuable and challenging scenarios are exercised first, while the unselected roads are retained in a lower-priority queue for potential future testing.

A. Curvature Analysis and Segmentation

Initial Test Suite: The initial test suite $T = \{T_i\}$ consists of tests composed of two main parts $T_i = (R_i, C_i)$, where R_i is the road and C_i is the configuration of the scenario that has to be simulated on the road R_i. A *road* is represented as an ordered sequence of Catmull-Rom spline points [20]:

$$R = \{p_1, p_2, \dots, p_n\}, \quad p_i = (x_i, y_i) \tag{1}$$

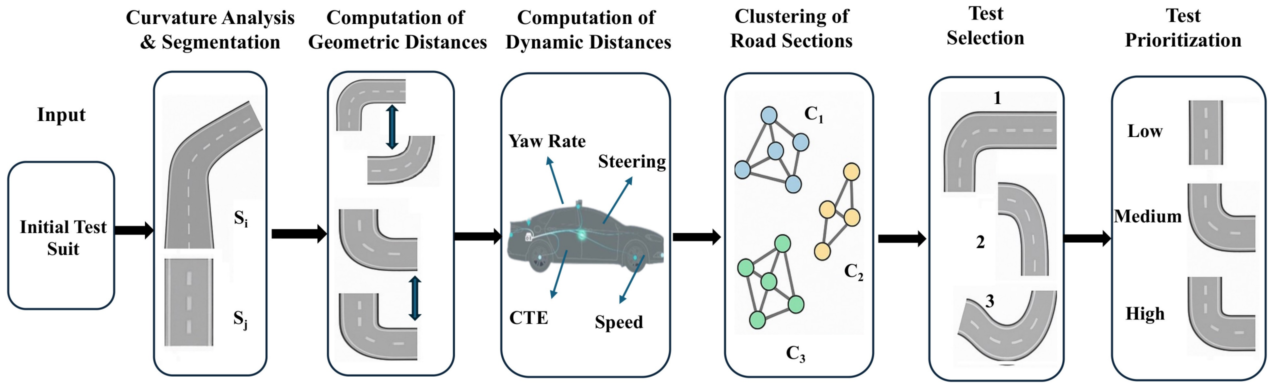

Curvature Analysis & Segmentation
Computation of Geometric Distances
Computation of Dynamic Distances
Clustering of Road Sections
Test Selection
Test Prioritization
Input
Initial Test Suit
S_i
S_j
Yaw Rate
Steering
CTE
Speed
C_1
C_2
C_3
1
2
3
Low
Medium
High

Fig. 1: Coverage-based road selection and prioritization framework.

where (x_i, y_i) are the coordinates along the road's centerline. The configuration C_i includes the information necessary to run the simulation, such as the initial position of the vehicle and its speed.

Curvature Calculation: For each road R, we calculate the curvature value at each point (x_i, y_i) along the centerline, generating a continuous sequence of curvature values κ_i that characterizes the road's geometry. Curvature plays a central role in our study, as it captures the essential shape of the road, indicating where it is straight (near-zero curvature values), where it bends (depending on the sign of the curvature value), and how sharply it turns (depending on the absolute curvature value). The curvature calculation employs a *three-point discrete approximation* based on the circumcircle radius of consecutive trajectory points. That is, for three consecutive trajectory points $p_{i-1} = (x_{i-1}, y_{i-1})$, $p_i = (x_i, y_i)$, $p_{i+1} = (x_{i+1}, y_{i+1})$, we first compute the determinant of the consecutive segment vectors to determine the turning direction:

$$\det = (x_i - x_{i-1})(y_{i+1} - y_i) - (y_i - y_{i-1})(x_{i+1} - x_i) \quad (2)$$

where $\det > 0$ corresponds to a counterclockwise (left) turn, while $\det < 0$ corresponds to a clockwise (right) turn, and $\det = 0$ indicates collinear points (straight line). Then the curvature magnitude *Rad* is obtained from the circumcircle radius of the three points, which is calculated using the formula [21]:

$$Rad = \frac{|p_i p_{i+1}| \cdot |p_{i-1} p_{i+1}| \cdot |p_{i-1} p_i|}{2|det|} \quad (3)$$

The actual curvature is $\kappa = 1/Rad$, where κ = 0 represents a straight line (when $\det = 0$, resulting in $Rad \to \infty$). This method captures both the sharpness of the curve (through curvature magnitude κ) and direction (through the sign) of road curves.

To divide roads into smaller, geometrically homogeneous sections for fine-grained focused testing (straight, left curve, right curve), we use segmentation. Segmentation enables fair comparison across roads; without it, comparing two entire roads directly would be misleading, since roads can vary greatly in length and overall shape. Moreover, segmentation highlights distinctive road features and creates a meaningful unit (road section, also called segment) for similarity matching, clustering, and prioritization.

Segmentation: The curvature-based segmentation process first employs a *hysteresis-based thresholding* framework [22] to categorize the shape of the road at each point by also considering the follow-up points, and then creates segments based on the computed information. In particular, given a sequence of w curvature values $\{\kappa_i, \kappa_{i+1}, \dots, \kappa_{i+w-1}\}$ starting at point p_i (i.e., κ_i is the curvature value of point p_i), the shape $s_i = \textit{shape}(p_i)$ of the road at position p_i is determined as follows.

- *straight*, if $\forall j = 1 \dots i + w - 1, |\kappa_j| < \tau_c$,
- *left-curve*, if $\forall j = 1 \dots i + w - 1, \kappa_j > \tau_c$,
- *right-curve*, if $\forall j = 1 \dots i + w - 1, \kappa_j < -\tau_c$,
- all other cases, retain the classification of the previous point.

where w is the length of the window, and τ_c is the threshold value for straight sections. We use a curvature threshold of $\tau_c = 0.015m^{-1}$ (corresponding to a $R = 66.67m$) since it has been reported to effectively distinguish between geometrically significant curves that require substantial steering input and nearly-straight sections that can be traversed with minimal control adjustments [23]. We use a hysteresis window $w = 3$ to reduce noise while maintaining responsiveness to genuine geometric transitions, following established principles for discrete curve analysis [22].

From a sequence of shape values s_i, with $s_i \in \{\textit{straight}, \textit{left-curve}, \textit{right-curve}\}$, segmentation has to establish the boundaries of each section. This is done by extracting subsequences of maximal length $\{s_j, \dots s_{j+k}\}$ with homogeneous shape, i.e., $s_j = s_{j+i} \forall i = 1 \dots k$ (e.g., a left-curve section where all its points are classified as left-curve). A minimum section length constraint of 10 meters is applied to avoid the generation of trivial sections. If a segment below the threshold is extracted, the segment is treated as a noisy section and merged into the preceding section, ensuring that each final section represents a meaningful geometric unit. The final output is a *sequence of non-overlapping sections* S_i whose union corresponds to the full road that has been segmented.

B. Computation of Geometric Distances

As a result of segmentation, we obtain many road sections, but not all of them are unique. Some sections may repeat across different roads (e.g., straight sections, identical curves) without contributing to exercising additional behaviors of the ADAS. We perform geometric matching to preserve distinctive geometries by identifying redundant sections and filtering out repeated patterns. In particular, we compare sections pairwise using Dynamic Time Warping (DTW) [19] on their curvature. Straight sections are flagged for coverage but excluded from similarity matching since they lack distinctive shape patterns.

DTW is a standard algorithm for measuring similarity between two sequences of potentially different lengths by performing non-linear temporal alignment, producing a distance (the difference between two curvature values) that reflects geometric similarity [19]. DTW ranges from 0 (identical sections) to 1 (completely different sections).

Since we look for redundancies in the set of roads to be exercised, we are not only interested in nearly-identical sections but also in the inclusion between segments. In fact, if a section is included into another one, the shorter section would represent a redundant scenario compared to the longer one. To capture this case, we distinguish how we compute the similarity between sections of similar lengths from the matching for inclusion between sections.

In particular, given two sections, P and Q, their length ratio is defined as $lr(P,Q) = \frac{min(|P|,|Q|)}{max(|P|,|Q|)}$. If $lr(P,Q) < \tau_{len}$, $sim(P,Q) = 1 - DTW(P,Q)$, that is, if the two sections have similar lengths, their similarity can be computed by using DTW directly. We empirically determined $\tau_{len} = 0.8$ as a good threshold to discriminate between sections that can be compared directly and sections that must be compared for inclusion.

If the two sections have different lengths (i.e., $lr(P,Q) \geq \tau_{len}$), the similarity between sections P and Q is computed by checking if the shortest section, namely P, is included in the longest one, namely Q. Note that if P is included in Q, using Q for testing implies having already tested the road geometry in P, but the opposite is not true.

The inclusion of P in Q is checked by considering every possible alignment between the two sections. In particular, for each alignment position k between 1 and $|Q| - |P|$, the similarity score of the considered alignment is defined as:

$$sim_k(P, Q[k : k + |P|]) = 1 - DTW(P, Q[k : k + |P|]) \quad (4)$$

where:

- $Q[k : k + |P|]$ denotes the subsequence of Q of length $|P|$ starting at index k.
- $|P|$ is the number of points in P.

Using a sliding-window framework, the shorter section P is aligned at every possible position k along the longer section Q, and DTW is computed for each alignment. The final similarity between P and Q is given by the best similarity value computed at various alignment points. This value represents the degree of inclusion of one road into the other, in terms of its geometry. More formally, this is defined as:

$$sim(P,Q) = max_{k=1\ldots|Q|-|P|} sim_k(P, Q[k : k + |P|]) \quad (5)$$

Finally, the distance function can be derived from the similarity is as follows: $d_{geom} = 1 - sim(P,Q)$.

C. Computation of Dynamic Distances

To better capture scenario difficulty beyond road geometry, we add information about vehicle dynamics to sections and use this information clustering, creating groups of sections that reflect similarity both in the geometry and in the driving behaviour. We extract and store dynamic data when tests are executed for the first time, so that they can be reused for test selection and prioritization when any model change occurs. In particular, we associate each section S with dynamic data $D(S) = \{di_i^S\}$ defined as follows: the speed variability $di_1^S = \sigma(\{v_t\}_{t \in S})$, the steering variability $di_2^S = \sigma(\{\theta_t\}_{t \in S})$, the mean cross-track error $di_3^S = \frac{1}{|S|}\sum_{t \in S} |cte_t|$, and the yaw rate variability $di_4^S = \sigma(\{\dot{\psi}_t\}_{t \in S})$, where:

- v_t = speed at time t,
- θ_t = steering angle at time t,
- cte_t = cross-track error at time t,
- $\dot{\psi}_t$ = yaw rate at time t,
- $\sigma(\cdot)$ = standard deviation operator.

As these metrics naturally operate on different scales, each feature is normalized to [0, 1] using min-max scaling to make them comparable. The four indicators capture information about the driving difficulty, the control complexity, the correctness, and the stability of the driving [24], [25].

We compute the distance among dynamic indicators of two sections P, Q as follows.

$$d_{\text{dyn}}(P,Q) = \frac{1}{4}\sum_{i=1}^{4}\left|di_i^P - di_i^Q\right| \quad (6)$$

This yields a distance value in [0, 1], with 0 indicating identical dynamic behavior and 1 maximal dissimilarity.

D. Clustering of Road Sections

After section matching and dynamic data integration, we perform clustering to group recurring road patterns based on both geometric and dynamic behavior similarity. This process identifies representative geometries, distinguishing common patterns from rare ones, and ensures that both typical and challenging driving scenarios are included in the test suite. By selecting a minimal set of roads that covers all clusters, we reduce testing redundant segments while maintaining comprehensive road coverage (as explained in Section III-E).

The clustering process uses an agglomerative clustering framework [26] organized by section type. Agglomerative clustering provides an optimal solution for road test generation by combining automatic cluster determination, hierarchical relationship modeling, deterministic reproducibility, and computational scalability [26], [27]. These advantages make it significantly superior to K-means' arbitrary partitioning and non-deterministic behavior [28], [29], as well as DBSCAN's inappropriate density assumptions and potential exclusion of safety-critical edge cases [30]. Sections are compared using a distance function that combines distances computed using geometric and dynamic data:

$$d_{hybrid}(S_i, S_j) = (1-w_{dyn})\cdot d_{geom}(S_i, S_j) + w_{dyn}\cdot d_{dyn}(S_i, S_j) \tag{7}$$

with $w_{dyn} \in [0,1]$ (default $w_{dyn} = 0.5$).

To identify the groups of similar road sections, we construct a pairwise distance matrix D using d_{hybrid}. We then apply agglomerative hierarchical clustering algorithm with *complete linkage* [26]. It begins by treating each section as an individual cluster, then it iteratively merges the closest clusters until an optimal result is reached (the algorithm automatically determines when to stop according to the distribution of the pairwise distances between elements).

E. Test Selection

Once clusters are defined, we identify the representative sections to be covered in each cluster to minimize redundancy while preserving geometric and dynamic diversity. Cluster representatives are used to define the coverage requirements, ensuring both typical and challenging road scenarios are included.

We distinguish two cases: singleton clusters and non-singleton clusters. For each singleton cluster $C = \{s\}$, we select as representative the only section included in C, that is $\mathit{Rep}(C) = s$. We refer to the set of all the representatives collected from singleton clusters with $\mathit{Rep}_{\mathit{singleton}}$. For each cluster with multiple sections, we use a diversity-driven framework to select up to three representatives per cluster. In particular, if the cluster includes three or fewer sections, we select all the sections. If more than three sections are available in a cluster, we compute the mean curvature of each section, and we select three representatives equally distributed across the spectrum of curvature values, intuitively selecting a section with low, medium and high curvature. The curvature $\bar{\kappa}(S)$ of a section S is computed as follows:

$$\bar{\kappa}(S) = \frac{1}{|S|}\sum_{i=1}^{|S|}\kappa_i \tag{8}$$

We indicate this set of representative sections as R_{multi}. The selected test cases aim to first cover the representative sections, that is, $\mathit{REP} = R_{\mathit{singleton}} \cup R_{\mathit{multi}}$. Once the representative sections REP have been identified, the set of tests T_{cov} that include all these sections is selected. That is, T_{cov} is initialized with the empty set, and then for each section $S \in R$, the test whose road includes S is added to T_{cov}.

The output of test selection thus split the initial test suite T into two test suites T_{cov}, which includes the high-priority test cases, and $T_{\mathit{surplus}} = T \setminus T_{\mathit{cov}}$, which includes the remaining low-priority test cases.

F. Tests Prioritization

Once T_{cov} and T_{surplus} have been defined, we determine the execution order of the tests inside each group using a multi-criteria prioritization framework that combines geometric complexity, dynamic behaviour metrics, and historical performance. That is, the tests that use roads with the most complex shape, produce the most challenging ADAS behaviour, and have already failed in past executions are executed first. To achieve this result, we introduce a test scoring mechanism, based on the combination of three scores.

Geometric score: Geometric scoring captures road shape diversity using the metrics curvature variation, number of high-curvature sections, and diversity of section types (left/right/straight). For each test $T = (R, C)$:

$$G(T) = w_{cv}\cdot\sigma(\mathit{curv}) + w_{hc}\cdot N_{hc} + w_{dt}\cdot D_{types} \tag{9}$$

where: $\sigma(\mathit{curv})$ is the standard deviation of curvature values (curvature variability), N_{hc} is the number of high-curvature sections in R exceeding the threshold $\kappa_{thr} = 0.015$ rad/m, and D_{types} is the number of distinct section types among {straight, left, right}$(D_{types} \in \{1,2,3\})$, with $w_{cv} = w_{hc} = w_{dt} = \frac{1}{3}$ for balanced geometric diversity assessment.

Dynamic score: Dynamic scoring quantifies scenario difficulty using the metrics speed variability, steering variability, cross-track error (cte) severity, and yaw rate variability. Given a test $T = (R, C)$, we compute the following indicators from its execution: the standard deviation of the speed of the vehicle σ_v, the standard deviation of the steering angle σ_θ, the mean absolute cross-track error cte, and the standard deviation of the angular velocity σ_ψ. The dynamic difficulty score is:

$$D(r) = \frac{1}{4}\left(\sigma_v + \sigma_\theta + \text{cte} + \sigma_\psi\right) \tag{10}$$

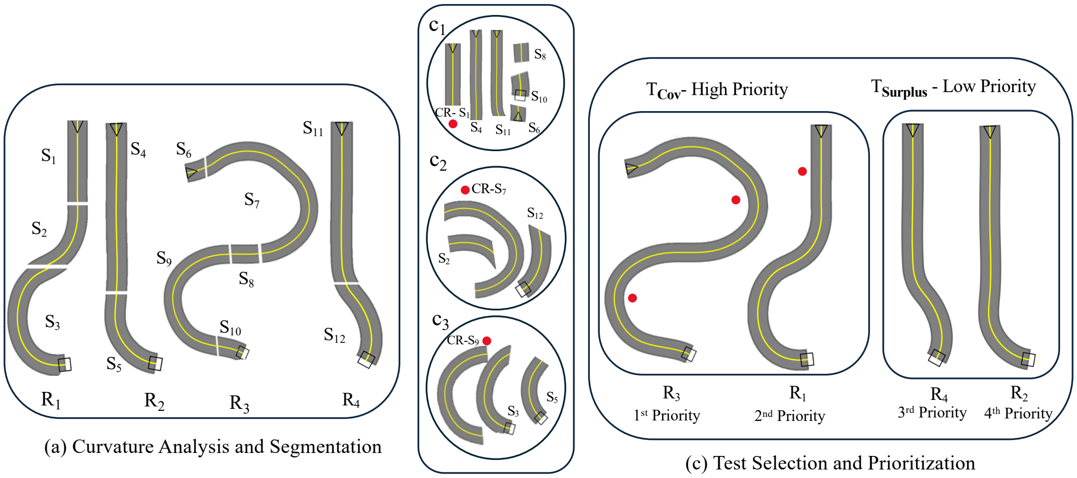
Historical performance score: Historical data $H(r)$ contributes by assigning extra points (0.25) to tests that previously caused simulation failures, ensuring regression testing focuses on known problematic cases, which is a well-known principle in regression testing [31], [32]. The final score is a weighted combination:

$$P(r) = \alpha\cdot G(r) + \beta\cdot D(r) + H(r) \tag{11}$$

where $\alpha + \beta = 1$, with default $\alpha = 0.5$ (geometric emphasis) and $\beta = 0.5$ (dynamic emphasis). All scores are normalized to $[0,1]$ before weighting to ensure comparability. The final execution order inside each group of tests is determined by descending values of $P(r)$. As a result, we get a ranked list of roads, each with a transparent breakdown of its contributing factors and rationale for its assigned priority.

Figure 2 provides a concrete illustration of the proposed framework, demonstrating the sequential operations described in Sections III-A–III-F.

In Part *(a)*, the input set of roads $R = \{R_1, R_2, R_3, R_4\}$ is decomposed into multiple sections $S = \{S_1, S_2, \ldots, S_{12}\}$ through curvature-based segmentation. This process captures



(a) Curvature Analysis and Segmentation

(b) Section Matching and Clustering

(c) Test Selection and Prioritization

Fig. 2: Illustrative example of the proposed framework

local geometric variations such as straight, left-curved, and right-curved sections, enabling fine-grained roads comparison.

Part *(b)* illustrates the section matching and clustering phase. Each section is compared against others based on its section type (straight vs. straight and so on), leading to the formation of clusters $C = \{C_1, C_2, C_3\}$. Each cluster groups similar sections and is represented by a *cluster representative* (CR), highlighted by a red dot in the figure. The CR serves as the most informative example of that section type (e.g., straight, a sharp curve, or a complex curvature pattern).

Finally, Part *(c)* presents the test selection and prioritization process. Roads containing one or more cluster representatives (e.g., R_3 and R_1) are selected to form the minimal yet diverse test suite, ensuring that the selected set covers the distinct geometric and dynamic behaviors identified across the dataset. These roads are assigned higher priority for testing (e.g., R_3 first and R_1 second), while the remaining roads (R_4 and R_2) are retained as lower priority candidates, providing supplementary coverage if additional testing resources permit.

This example illustrates how the proposed framework systematically decomposes complex road networks into comparable structural units, clusters them to identify representative geometric and dynamic behaviors, and subsequently prioritizes test scenarios based on their diversity and criticality.

IV. Empirical Study

The goal of our empirical study is to evaluate the effectiveness of our test selection and prioritization methods in reducing redundancy and improving efficiency. Specifically, we assess how well the combination of geometric segmentation, dynamic data integration, and historical failure information minimizes the number of test roads while preserving scenario diversity and enabling effective test prioritization.

A. Research Questions

RQ$_1$ (Selection): *How effective is the selected test suite?*

This research question investigates the cost-effectiveness of the test suite obtained by selecting the tests that cover the clusters' representatives. A good test suite must discover a good number of issues using a few tests only.

RQ$_2$ (Effectiveness of Dynamic Data): *To what extent does considering dynamic driving behavior enhance the quality of test selection and the effectiveness of fault detection compared to the geometry-only method?*

This research question evaluates the individual impact of static data (i.e., road geometry) and dynamic data (i.e., the driving behavior) on the overall effectiveness of the test selection and prioritization processes.

RQ$_3$ (Prioritization): *How effective is the proposed prioritization strategy in identifying critical scenarios and detecting failures?*

This research question investigates whether the proposed prioritization method can schedule test executions in a way that critical scenarios are likely to be executed early, compared to a random order of the tests.

RQ$_4$ (Cross-Model Generalization): *How effective is test prioritization in revealing failures across different autonomous driving models?*

This RQ investigates the effectiveness of the prioritization across different model architectures, studying whether the features considered by our approach can generalize across different ADAS models.

B. Objects of Study

To assess test selection and prioritization, we consider test suites exercising NHTSA [33] Level 2 ADAS, which perform vision-based perception tasks using data gathered by the camera sensors of a vehicle. Despite the adoption of Level 2 ADAS in many commercial vehicles, their reliability remains a concern, as evidenced by numerous recent crash reports [34] and real-world validation experiments [35]. Although Levels 3 and 4 ADAS have been proposed [36], their real-world deployment remains highly constrained. Consequently, addressing the limitations of Level 2 systems is crucial for advancing to higher levels of autonomy. We focus on a specific ADAS application, that is, a system for lane-keeping assistance (LKA).

As a model architecture to address the LKA task we consider DAVE-2, which is a convolutional neural network developed for multi-output regression tasks based on imitation learning [37]. The model architecture includes three convolutional layers for feature extraction, followed by five fully connected layers. DAVE-2 has been extensively used in a variety of ADAS testing studies [38], [39], [40], [41], [42], [43]. The model takes as input an image representing a road scene, and it is trained to predict the vehicle's actuator commands. Our implementation includes a DNN with lane-keeping (LK) and adaptive cruise control (ACC) capabilities, as DAVE-2 is trained to conduct the vehicle on the right lane of the road at the maximum possible speed, by predicting appropriate steering and throttle commands.

C. Experimental Platforms and Benchmarks

We conducted experiments on the Udacity [44] simulation environment since it is open-source and suitable for Level 2 ADAS evaluation. Simulation platforms are widely used for testing of ADAS, as researchers have shown that model-level testing is inadequate at exposing system-level failures [14], [45], [16]. Udacity [44] is developed with Unity 3D [46], a popular cross-platform game engine, based on the Nvidia PhysX engine [47], featuring discrete and continuous collision detection, ray-casting, and rigid-body dynamics simulation.

As a benchmark, we used the test scenarios available in the OPENCAT dataset [20], which provides 32,580 different road scenarios converted from the SENSODAT benchmark [7] across three campaigns: *Ambiegen, Frenetic, and Frenetic_v.*

D. Procedure and Metrics

RQ₁ (Selection): We evaluate the effectiveness of coverage-based test selection in comparison to random test ordering using the following metrics:

Reduction Ratio (%): The reduction ratio quantifies the efficiency of the test selection process by measuring the percentage of tests eliminated from the original test suite. A higher ratio indicates more efficient reduction. It is defined as:

$$\text{Reduction Ratio (\%)} = \frac{|T_{\text{surplus}}|}{|T|} \times 100 \qquad (12)$$

Failed Roads Retention (%): This metric measures the effectiveness of the reduction by considering the percentage of failed tests in the selection. It is defined as:

$$\text{Failed Roads Retention (\%)} = \frac{|T_{cov} \cap F|}{|F|} \times 100 \qquad (13)$$

where F represents the set of failed tests, and T_{cov} is the selected set. We consider the average number of failed roads discovered by a random selection of N tests, for multiple values of N, as a baseline.

RQ₂ (Effectiveness of Dynamic Data): To investigate this question, we compare two configurations of the selection pipeline: *Geometric-only selection (Geo-Only)*, which clusters and selects tests only using information about the geometry of the road, as described in Section III-B; and *Hybrid (geometric + dynamic)*, which uses both geometric and dynamic features as described in Section III-C.

Both configurations are applied to the same set of road scenarios across the campaigns, and results are compared in terms of their ability to retain historically failed roads within the reduced subset. We use the same metrics used for RQ to evaluate the effectiveness of the test selection and prioritization. In addition, we measure the relative improvement that dynamic data brings to the use of static data only.

RQ₃ (Prioritization): We use the following metrics to evaluate the performance of the prioritization approach.

Early Fault Detection (EFD): This metric measures the percentage of failed tests selected among the first k tests in the prioritized list. We use this metric to evaluate early fault detection capability, quantifying how quickly our approach identifies safety-critical scenarios compared to random ordering. We consider two cases, $k = 10$ to assess the capability to observe misbehaviors immediately after the test suite is executed, and $k = |T_{cov}|$ (which is the same number of tests selected by our approach), to assess how good the initial part of the test suite is. When considering random ordering, we compute the actual average percentage of failed tests that occur in the first k tests.

APFD (Average Percentage of Fault Detection) Calculation: We compute how good a test ordering is using the well-established APFD [48] metric:

$$APFD = 1 - \frac{\sum_{i=1}^{m} TF_i}{n \times m} + \frac{1}{2n}$$

where TF_i represents the position of the first test revealing fault i, n is the total number of tests (roads), and m is the total number of failures. Higher APFD values indicate better fault detection across the test execution sequence.

RQ₄ (Cross-Model Generalization): To evaluate cross-model generalization, we used the same set of tests obtained from *RQ₁ (Selection)* but upgraded the ADAS model from Dave-2 to the Chauffeur model [49]. The two models differ significantly in their design and represent architecturally distinct approaches to autonomous driving. DAVE-2 is a feedforward network with five convolutional and three fully connected layers [37], while Chauffeur employs a deeper convolutional structure (six layers with dropout and pooling) and critically integrates recurrent LSTM layers to process temporal context. While DAVE-2 is a larger model, Chauffeur's incorporation of memory through its recurrent design provides a strong basis for evaluating the generalization of our road scenarios across fundamentally different model architectures.

We investigate the capability of the selected test cases to reveal failures also for Chauffeur, analyzing the degree of consistency observed for the tests failed by Chauffeur (F_C) and DAVE-2 (F_D).

E. Results

This section presents the results of our empirical evaluation. Table I summarizes the outcomes of the coverage-based road selection and prioritization experiments conducted on a total of

TABLE I: RQ_1: Coverage-based road selection and prioritization.

Campaign	Total No. Tests	No. Failed Tests	Selected Tests	Reduction %	FRR Selected%	EFD RnD %	EFD10 Tests	EFD10 Rnd Tests	APFD
Ambiegen Campaigns									
2	973	11	147	85%	45%	0.17%	45%	1.04%	0.92
3	964	9	206	79%	89%	0.20%	80%	1.04%	0.95
4	965	5	178	82%	80%	0.10%	80%	1.04%	0.93
5	958	10	167	83%	80%	0.18%	70%	1.04%	0.91
6	959	9	179	81%	78%	0.18%	70%	1.04%	0.89
7	963	10	197	80%	70%	0.21%	60%	1.04%	0.96
8	952	11	176	82%	91%	0.21%	91%	1.05%	0.92
9	953	4	187	80%	100%	0.08%	75%	1.05%	0.97
10	971	18	176	82%	89%	0.34%	56%	1.03%	0.85
11	973	10	190	80%	80%	0.20%	80%	1.03%	0.93
13	954	7	185	81%	86%	0.14%	86%	1.05%	0.94
14	959	8	187	80%	75%	0.16%	75%	1.05%	0.82
15	952	19	206	78%	63%	0.43%	53%	1.05%	0.96
Frenetic Campaigns									
2	928	7	27	97%	57%	0.02%	57%	1.08%	0.88
3	954	11	41	96%	73%	0.05%	73%	1.06%	0.91
4	964	12	29	97%	58%	0.04%	58%	1.05%	0.92
5	945	8	30	97%	63%	0.03%	75%	1.06%	0.93
6	944	16	33	97%	44%	0.06%	44%	1.06%	0.90
7	967	14	38	96%	57%	0.03%	57%	1.03%	0.97
8	952	10	30	97%	60%	0.03%	70%	1.05%	0.86
9	964	6	37	97%	67%	0.04%	67%	1.04%	0.97
11	866	11	37	96%	64%	0.05%	64%	1.15%	0.94
12	956	17	39	96%	59%	0.07%	59%	1.05%	0.89
13	959	13	34	96%	54%	0.05%	54%	1.04%	0.95
14	866	11	33	96%	73%	0.05%	73%	1.15%	0.97
15	870	12	37	96%	67%	0.06%	67%	1.15%	0.85
Frenetic_v Campaigns									
2	944	7	31	97%	86%	0.02%	87%	1.06%	0.92
4	525	3	25	95%	67%	0.03%	67%	1.90%	0.87
5	940	7	21	98%	100%	0.02%	100%	1.06%	0.95
6	764	5	22	97%	100%	0.02%	100%	1.31%	0.92
7	47	0	8	83%	-	1.45%	-	-	-
11	953	8	33	97%	88%	0.03%	88%	1.05%	0.90
12	942	7	27	97%	71%	0.02%	71%	1.06%	0.87
13	951	13	34	96%	54%	0.03%	87%	1.05%	0.83
14	934	9	35	96%	89%	0.04%	89%	1.07%	0.90
15	949	7	33	97%	86%	0.03%	86%	1.05%	0.85

32,580 test roads across the three OpenCat [20] campaigns: *Ambiegen*, *Frenetic*, and *Frenetic_v*. Columns *Campaign*, *total No. Tests* and *No. Failed Tests* indicate the id of the campaign, the number of tests available in that campaign and the total number of tests failed by DAVE-2 in the campaign, respectively. Columns *Selected* and *Reduction* indicate the absolute number and percentage of tests selected by our approach (i.e., the tests in $|T_{cov}|$), respectively. Column *FRR Selected %* reports the percentage of failures that are revealed by the selected test cases. Column *EFD RnD* indicates the average percentage of faults discovered by randomly selecting as many tests as the ones in $|T_{cov}|$. Columns *EFD10 Tests* and *EFD10 Rnd Tests* show the percentage of failures revealed by the first 10 test cases selected with our approach and randomly, respectively. Finally, column *APFD* indicates the Average Percentage of Fault Detection metric for the prioritized test suite.

RQ₁ (Selection): The results demonstrate that our approach substantially reduces the number of roads while maintaining strong representational coverage of behavioral and geometric diversity. In the *Ambiegen* campaign, the number of selected roads decreased by an average of 81% compared to the original dataset, while preserving between 70–100% of the failed roads. Similar trends were observed for the *Frenetic* and *Frenetic_v* campaigns, achieving average reductions of 96% and 94%, respectively. Despite this drastic reduction, the selection maintained coverage of at least 57–100% of failed roads, indicating that the clustering-based selection retained roads representing critical geometric and dynamic behaviors, which can be used to discover several failures quickly. Such an early discovery of failures can then be backed up by the execution of the remaining prioritized tests to reveal any possible remaining failure, according to the resources available and development strategy (e.g., first fixing the failures revealed by the prioritized tests before running the remaining tests).

RQ₂ (Effectiveness of Dynamic Data): Figure 3 illustrates the comparative results of test selection (left plot) and fault detection rate (right plot) using road geometry only (*Geo-only*) and including dynamic data (*w/Dyn*). Integrating dynamic behavior data led to substantial improvements in test selection (coverage) and fault detection effectiveness across all campaigns, at the cost of a light increase in the number of selected tests. The slight increase in the selected tests reflects

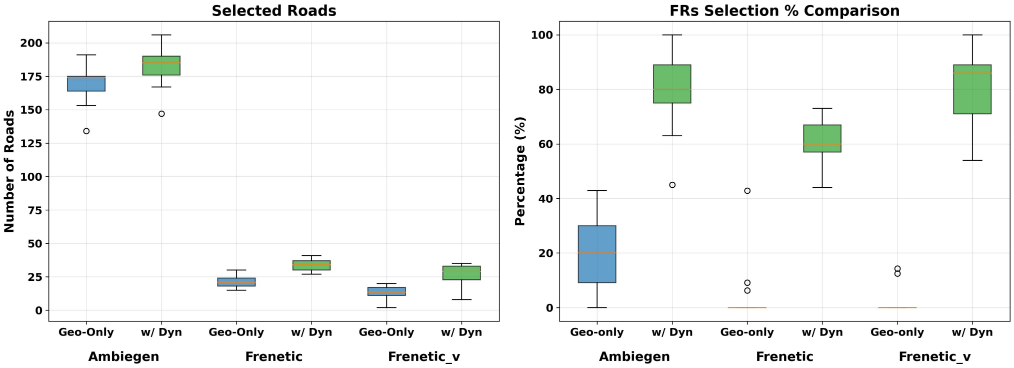


Fig. 3: Comparison between Geometric-only and Hybrid approaches.

the additional diversity factors introduced by dynamic data. In particular, for the *Ambiegen* campaigns, the average number of selected tests increases from 170 tests (geometric-only) to 185 tests (hybrid) when dynamic attributes were considered, with a corresponding improvement of the fault detection rate from 18% to 82%. Similarly, in the *Frenetic* and *Frenetic_v* campaigns, the average number of selected roads increases from 20 and 18 to 35 and 32, respectively. Correspondingly, the fault detection rate increases from 5% and 3% to 70% and 65%, respectively. These results demonstrate that dynamic data is a critical component for test selection.

RQ_3 (Prioritization): Table I reports the *Early Fault Detection (EFD)* for the selected test suite (column FRR Selected %) and the same number of randomly selected test cases (EFD Rnd). Moreover, it reports the same metric when only 10 test cases are selected. This is to study how effective the very top selected test cases are in revealing potential issues in the ADS under test. Specifically, in the *Ambiegen* campaign, the hybrid approach achieved EFD-Top10 rates between 70% and 86%, surpassing the baseline ($\approx$ 1.04–1.05%). The *Frenetic* and *Frenetic_v* campaigns exhibited similar trends, where the hybrid method reached up to 87–100%, while the normalized baseline remained near the statistical expectation ($\approx$ 1.05–1.15%). These results demonstrate that the hybrid approach successfully prioritizes the most fault-prone roads early in the testing sequence. Across all campaigns, we achieved approximately 60–90× higher early fault detection efficiency in the hybrid prioritization than the statistical baseline, confirming its effectiveness in ranking safety-critical roads.

In addition to EFD, we computed the *Average Percentage of Fault Detection (APFD)* to evaluate the overall prioritization efficiency across the complete test suite. The $APFD$ results range from 0.82 and 0.97 across all campaigns, reflecting the fraction of cumulative fault detection achieved as testing progresses. These values indicate that, on average, 85–97% of total faults are detected within the first half of the prioritized execution order, demonstrating that faults are concentrated toward the beginning of the sequence rather than uniformly across it. This trend reinforces that the hybrid prioritization approach effectively concentrates high-risk, fault-prone roads early in the execution process, thereby maximizing detection efficiency and resource utilization.

RQ_4 (Cross-Model Generalization): Figure 4 and Figure 5 illustrate the results for the cross-model generalization *RQ_4*.

The evaluation revealed quite a different behavior for the selected tests once executed with DAVE-2 compared to Chauffeur, with Chauffeur failing fewer tests than DAVE-2 in the scope of T_{cov}. This is particularly true for the *Frenetic* and *Frenetic_v* campaigns, where the selected tests failed by Chauffeur are a subset of the ones failures by DAVE-2. While in *Ambiegen* campaigns, Chauffeur failed tests that DAVE-2 passed. Yet, also in this campaign, DAVE-2 failed more tests than Chauffeur, in the scope of T_{cov}.

Overall, the findings through *RQ_1-RQ_4* validate the effectiveness of integrating geometric complexity, dynamic variation, and historical failure information in identifying representative and diverse roads while minimizing redundant tests. The consistent inclusion of most failure-prone roads suggests that the method captures the spectrum of safety-critical geometries and dynamic driving conditions, producing a more discriminative prioritization ranking. Specifically, the findings demonstrate that the proposed selection strategy preserves geometric and behavioral coverage (*$RQ_{1,2}$*), the hybrid prioritization significantly improves early fault detection when a same model is used (*RQ_3*), and the selected roads have the potential to early reveal failures also in other models (*RQ_4*). Collectively, these results suggest that the proposed prioritization strategy may generate cost-effective prioritized aware test suites for autonomous driving evaluation.

F. Threats to Validity

Regarding internal validity, our framework relies on a combination of geometric and dynamic features to guide test selection and prioritization. While the thresholds and weights used were selected based on prior literature and default values, alternative configurations may yield different results. Nevertheless, the consistent performance across multiple campaigns and models suggests that the chosen parameters are robust.

External validity is limited by the use of the OPENCAT dataset and the Udacity simulator, which offer a rich and diverse

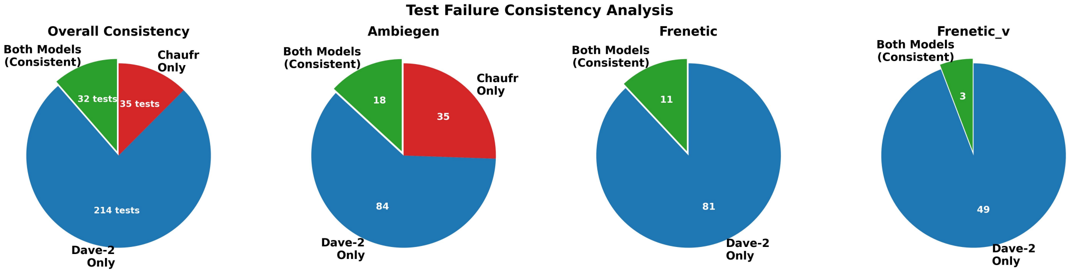


Fig. 4: Cross-model test failure consistency analysis for the selected tests.

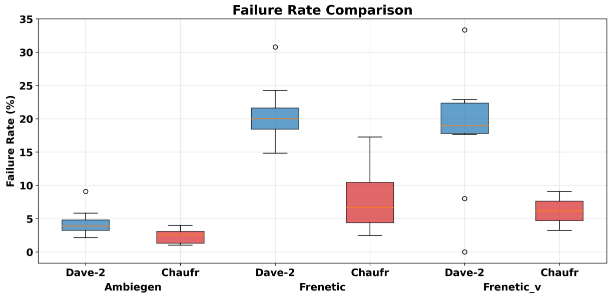


Fig. 5: Cross-Model Failure Distribution (Dave-2/Chauffeur).

set of road scenarios. While these platforms provide a controlled and scalable environment for testing, they may not fully capture the complexity of real-world driving conditions, such as sensor noise, unpredictable traffic, or weather variability. However, the ability to systematically evaluate thousands of scenarios across different campaigns strengthens the generalizability of our approach, and future work will extend validation to physical-world platforms and additional datasets.

Construct validity may be affected by the metrics and features chosen to represent driving behavior and scenario diversity. Although DTW and the selected dynamic features aim to capture meaningful aspects of driving complexity, they may not fully encompass all factors contributing to failure-inducing scenarios, such as rare edge cases or specific driving behaviors. To mitigate this, our study includes multiple metrics and validation across different models and datasets, but further refinement of feature representation and incorporation of real-world noise and variability remains an important direction for future research.

V. Related work

ADAS testing faces significant scalability challenges due to the large number of possible driving scenarios and the safety-critical nature of these systems [50], [51]. Traditional exhaustive testing approaches are computationally prohibitive, leading researchers to develop intelligent test reduction and prioritization strategies.

Classical test reduction and prioritization techniques focus on maximizing fault detection rates using coverage-based, fault-based, or requirement-based approaches to select or rank test cases [52]. However, in the context of autonomous driving systems, where the scenario space is vast and highly variable, classical techniques may not sufficiently capture the diversity of real-world behaviors. To address this, recent research has focused on scenario-based test reduction and prioritization methods that aim to minimize redundancy while preserving scenario diversity and fault detection capability [53], [54], [55], [56], [6]. Deng et al. [53] propose scenario-based test reduction and prioritization that segments driving recordings into semantically homogeneous scenes using a custom schema that encodes dynamic and static features. Redundant segments are removed based on vector similarity, and the remaining ones are prioritized using a combination of semantic coverage and rarity heuristics. Clustering techniques have also been explored to reduce redundancy and improve coverage. Kerber et al. [54] introduce a hierarchical agglomerative clustering approach based on a custom scenario distance metric derived from spatiotemporal features of highway driving data, enabling the grouping of similar scenarios and supporting data-driven estimation of test coverage. Bernhard et al. [55] propose a multi-stage trajectory clustering method that combines Gaussian Mixture Models and hierarchical clustering to group vehicle interactions based on spatial and directional behavior. Lu et al. [56] propose SPECTER, a search-based approach to select and prioritize test scenarios in autonomous driving systems. Their method uses multi-objective evolutionary algorithms to balance competing objectives, including scenario diversity, criticality, and execution cost. The approach demonstrates significant improvements over random and greedy baseline methods when applied to large scenario repositories. Birchler et al. [6] propose SDC-Prioritizer, which uses static road features and meta-heuristic algorithms to prioritize test cases for self-driving cars. Their approach demonstrates the effectiveness of combining multiple geometric and topological features to rank test cases, achieving significant improvements in the detection of safety-critical failures.

Current approaches typically focus on geometric or behavioral characteristics in isolation, missing opportunities for comprehensive scenario characterization. Our work addresses this limitation by integrating geometric similarity assessment with behavioral analysis using dynamic driving data, while providing multi-criteria prioritization that combines geometric complexity, behavioral metrics, and historical test data.

VI. Conclusions

In this paper, we introduced a framework for test selection and prioritization in ADAS, designed to improve testing efficiency while preserving behavioral and geometric diversity. By segmenting road scenarios, incorporating dynamic driving behavior, and clustering to remove redundancy, the framework selects and orders tests based on their potential to expose failures. In our experiments on the OPENCAT dataset with two ADAS models, our approach reduced test suite size by up to 89% while retaining an average of 73% of failing cases. The prioritization strategy accelerated fault detection by up to 95× over random baselines, achieving APFD scores between 0.82 and 0.97, consistently exposing critical failures early. These results demonstrate that the method provides a practical balance between efficiency and robustness, making it suitable for cost-effective ADAS regression testing.

VII. Data Availability

The pipeline used to obtain the results discussed in this work and the results are available in our replication package [9].

References

[1] V. G. Cerf, "A comprehensive self-driving car test," *Commun. ACM*, vol. 61, no. 2, pp. 7–7, Jan. 2018. [Online]. Available: http://doi.acm.org/10.1145/3177753

[2] Zinnov, "Automotive outsourcing analysis," https://zinnov.com/wp-content/themes/zinnov/images/illustrative-overview/reports/Automotive-Outsourcing-Analysis.pdf, 2018.

[3] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[4] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[5] K. van Wees, "Vehicle safety regulations and adas: tensions between law and technology," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 4, 2004, pp. 4011–4016 vol.4.

[6] C. Birchler, S. Khatiri, P. Derakhshanfar, S. Panichella, and A. Panichella, "Single and multi-objective test cases prioritization for self-driving cars in virtual environments," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 2, Apr. 2023. [Online]. Available: https://doi.org/10.1145/3533818

[7] C. Birchler, C. Rohrbach, T. Kehrer, and S. Panichella, "Sensodat: Simulation-based sensor dataset of self-driving cars," in *Proceedings of the 21st International Conference on Mining Software Repositories*, ser. MSR '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 510–514. [Online]. Available: https://doi.org/10.1145/3643991.3644891

[8] A. Gambi, M. Mueller, and G. Fraser, "Automatically testing self-driving cars with search-based procedural content generation," in *Proceedings of the ACM 28th SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 318–328.

[9] Github, "Replication package," 2025, accessed: 2025-10-15. [Online]. Available: https://github.com/testeraxion/prioritization

[10] BGR Media, LLC, "Waymo's self-driving cars hit 10 million miles," https://techcrunch.com/2018/10/10/waymos-self-driving-cars-hit-10-million-miles, 2018, online; accessed 18 August 2024.

[11] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, "Testing Machine Learning based Systems: A Systematic Mapping," *Empirical Software Engineering*, 2020.

[12] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of Real Faults in Deep Learning Systems," in *Proceedings of 42nd International Conference on Software Engineering*, ser. ICSE'20. ACM, 2020, p. 12 pages.

[13] A. Stocco, M. Weiss, M. Calzana, and P. Tonella, "Misbehaviour prediction for autonomous driving systems," in *Proceedings of ACM 42nd International Conference on Software Engineering*, ser. ICSE '20. ACM, 2020, p. 12 pages.

[14] F. U. Haq, D. Shin, S. Nejati, and L. Briand, "Comparing offline and online testing of deep neural networks: An autonomous car case study," in *Proceedings of 13th IEEE International Conference on Software Testing, Verification and Validation*, ser. ICST '20. IEEE, 2020.

[15] F. U. Haq, D. Shin, and L. C. Briand, "Efficient online testing for dnn-enabled systems using surrogate-assisted and many-objective optimization," in *44th IEEE/ACM 44th International Conference on Software Engineering, ICSE 2022, USA*. ACM, 2022, pp. 811–822.

[16] A. Stocco, B. Pulfer, and P. Tonella, "Model vs system level testing of autonomous driving systems: a replication and extension study," *Empirical Software Engineering*, vol. 28, no. 3, p. 73, May 2023.

[17] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Softw. Test. Verif. Reliab.*, vol. 22, no. 2, p. 67–120, Mar. 2012. [Online]. Available: https://doi.org/10.1002/stv.430

[18] A. Vahabzadeh, A. Stocco, and A. Mesbah, "Fine-grained test minimization," in *Proceedings of the 40th ACM/IEEE International Conference on Software Engineering*, ser. ICSE 2018. ACM, may 2018, pp. 210–221.

[19] M. Müller, *Dynamic Time Warping*. Springer Berlin Heidelberg, 2007, pp. 69–84. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4

[20] Q. Ali, A. Stocco, L. Mariani, and O. Riganelli, "OpenCat: Improving Interoperability of ADS Testing," in *Proceedings of 47th International Conference on Software Engineering Workshops*, ser. ICSEW '24. IEEE, 2025, p. 10 pages, dataset available at https://github.com/lakhanqurban/OpenCat.

[21] P. Malgouyres and F. Brunet, "Digital curvature estimation based on osculating circle," *Machine Vision and Applications*, vol. 18, no. 3-4, pp. 229–251, 2007.

[22] J. Gielis, "A generic geometric transformation that unifies a wide range of natural and abstract shapes," *American Journal of Botany*, vol. 90, no. 3, pp. 333–338, 2003.

[23] American Association of State Highway and Transportation Officials, *A Policy on Geometric Design of Highways and Streets*, 7th ed. Washington, DC: American Association of State Highway and Transportation Officials, 2018.

[24] J. M. Anderson, N. Kalra, K. D. Stanley, P. Sorensen, C. Samaras, and O. A. Oluwatola, *Autonomous Vehicle Technology: A Guide for Policymakers*. Santa Monica, CA: RAND Corporation, 2016.

[25] D. Zhao, H. Peng, T. Lam, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles in lane-change scenarios based on importance sampling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2016.

[26] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[27] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[28] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," pp. 1027–1035, 2007.

[29] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," pp. 226–231, 1996.

[31] G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold, "Prioritizing test cases for regression testing," *IEEE Transactions on software engineering*, vol. 27, no. 10, pp. 929–948, 2001.

[32] L. Zhang, D. Marinov, L. Zhang, and S. Khurshid, "An empirical study of junit test-suite reduction," in *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2013, pp. 398–407.

[33] U. D. of Transportation, "A framework for automated driving system testable cases and scenarios," https://rosap.ntl.bts.gov/view/dot/38824/dot_38824_DS1.pdf, 2018.

[34] ——, "Standing general order on crash reporting for level 2 advanced driver assistance systems," https://www.nhtsa.gov/sites/nhtsa.gov/files/2022-06/ADAS-L2-SGO-Report-June-2022.pdf, 2022.

[35] J. Opletal. (2025, Jul.) China's massive adas test: 36 cars, 15 hazard scenarios, 216 crashes. [Online]. Available: https://carnewschina.com/2025/07/24/chinas-massive-adas-test-36-cars-15-hazard-scenarios-216-crashes/

[36] Baidu Inc., "Baidu Apolloscapes Dataset," https://apolloscape.auto/index.html, 2018, accessed: [2024-01-15].

[37] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars." *CoRR*, vol. abs/1604.07316, 2016.

[38] A. Stocco, B. Pulfer, and P. Tonella, "Mind the Gap! A Study on the Transferability of Virtual Versus Physical-World Testing of Autonomous Driving Systems," *IEEE Transactions on Software Engineering*, vol. 49, no. 04, pp. 1928–1940, apr 2023.

[39] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the ACM 40th International Conference on Software Engineering*, ser. ICSE '18. ACM, 2018, p. 303–314.

[40] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems," in *Proceedings of the IEEE/ACM 33rd ASE*, ser. ASE '18. ACM, 2018, p. 132–142.

[41] M. Biagiola, A. Stocco, V. Riccio, and P. Tonella, "Two is better than one: digital siblings to improve autonomous driving testing," *Empirical Softw. Engg.*, vol. 29, no. 4, May 2024.

[42] G. Jahangirova, A. Stocco, and P. Tonella, "Quality metrics and oracles for autonomous vehicles testing," in *Proceedings of the IEEE 14th International Conference on Software Testing, Verification and Validation*, ser. ICST '21. IEEE, 2021.

[43] M. Biagiola and P. Tonella, "Boundary state generation for testing and improvement of autonomous driving systems," *IEEE Trans. Softw. Eng.*, vol. 50, no. 8, p. 2040–2053, Jul. 2024. [Online]. Available: https://doi.org/10.1109/TSE.2024.3420816

[44] Udacity, "Udacity self-driving car simulator," https://github.com/udacity/self-driving-car-sim, 2021, accessed: [2024-01-15].

[45] F. U. Haq, D. Shin, S. Nejati, and L. Briand, "Can offline testing of deep neural networks replace their online testing? a case study of automated driving systems," *Empirical Software Engineering*, vol. 26, no. 5, p. 90, 2021.

[46] "Unity," https://unity.com/, 2024, accessed: 11-01-2024.

[47] "Nvidia PhysX," https://developer.nvidia.com/physx-sdk, 2022.

[48] G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold, "Test case prioritization: an empirical study," *Proceedings IEEE International Conference on Software Maintenance - 1999 (ICSM'99). 'Software Maintenance for Business Change' (Cat. No.99CB36360)*, pp. 179–188, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:15621196

[49] Team Chauffeur, "Steering angle model: Chauffeur," https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/chauffeur, 2016, accessed: [14-10-2025]. [Online]. Available: https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/chauffeur

[50] G. Lou, Y. Deng, X. Zheng, M. Zhang, and T. Zhang, "Testing of autonomous driving systems: where are we and where should we go?" in *In Proceedings of the Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2022.

[51] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y.-F. Li, L. Ma, Y. Xue, and Y. Liu, "A survey on automated driving system testing: Landscapes and trends," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 5, Jul. 2023. [Online]. Available: https://doi.org/10.1145/3579642

[52] S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Software testing, verification and reliability*, vol. 22, no. 2, pp. 67–120, 2012.

[53] Y. Deng, X. Zheng, M. Zhang, G. Lou, and T. Zhang, "Scenario-based test reduction and prioritization for multi-module autonomous driving systems," in *In Proceedings of the Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, Singapore, Singapore, 2022.

[54] J. Kerber, S. Wagner, K. Groh, D. Notz, T. Kühbeck, D. Watzenig, and A. Knoll, "Clustering of the scenario space for the assessment of automated driving," in *In Proceedings of the Intelligent Vehicles Symposium (IV)*, Aachen, Germany, 2020.

[55] J. Bernhard, M. Schutera, and E. Sax, "Optimizing test-set diversity: Trajectory clustering for scenario-based testing of automated driving systems," in *In Proceedings of the International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, US, 2021.

[56] C. Lu, H. Zhang, T. Yue, and S. Ali, "Search-based selection and prioritization of test scenarios for autonomous driving systems," in *In Proceedings of the International Symposium on Search Based Software Engineering (SSBSE)*, Bari, Italy, 2021.